ORIE 5270 - Assignment 0

You are not allowed to collaborate with your classmates, but you may freely use the internet to look up Python documentation that may help you! At the start of each file, give citations to any resources used.This homework is optional. If you submit it, it will count as a bonus homework, i.e. we will drop your worst grade and substitute it with this one.

One of the few prerequisites of ORIE5270 is basic knowledge of Python. The goal of this homework is for you to get familiar with this programming language, at least to the level we expect you to have in this class. If you don't feel comfortable with it, I recommend you check one of the LinkedIn Learning classes provided by Cornell, for example: https://www.linkedin.com/learning/learning-python-2/welcome?u=76816458.

Setup

Gradient descent is a common iterative method use to solve problems like

$$\min_x f(x)$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth function. The method generates a sequence of points $x_0, x_1, \ldots, x_T$ by applying a simple loop based on the following update:

$$x_{k+1} \leftarrow x_k - \eta \nabla f(x_k)$$

where η is a fixed parameter known as step size (or learning rate) and $\nabla f(x_k)$ denotes the gradient of f at x_k.

Problem 1

Implement a script called *problem1.py* that executes gradient descent for the function $f(x) = x^2 - x$. Set the initial point to $x_0 = 0$, the step size to $\eta = 0.5$ and the maximum number of iterations to $T = 100$. Your code must print a line every 10 iterations with the following format:

```
Iteration <k>, Objective value <obj>, Iterate <xk>
```

where `<k>`, `<obj>` and `<xk>` must be substituted with the values at the current iteration.

Problem 2

Implement a function with the following signature

```
def gradient_descent(x0, objective, gradient, step_size = 0.001,
     max_iter=100, print_frequency=10):
```

where `x0` is a numpy vector, and `objective` and `gradient` are functions with signature:

```
def objective(x):
  ...

def gradient(x):
  ...
```

and evaluate to the objective value and the gradient at `x`, respectively. Again, make sure the `gradient_descent` function prints a line with the objective value every `print_frequency` iterations.

Problem 3

Implement a script named *problem3.py* that applies the function you implemented in the previous problem to minimize the least squares objective, that is

$$\min_x \frac{1}{2}\|Ax - b\|^2$$

where A and b are given.

To set the stage copy the following block at the beggining of your script:

```
import numpy as np

A = np.random.randn(100, 50)
x = np.ones(50)
b = A.dot(x) + 0.01*np.random.randn(100)
step_size = 0.01
T = 1000
```

Use this to implement the functions `objective(x)` and `gradient(x)` for this problem. Then call the `gradient_descent(...)` function with this input. Use the output to make a plot of the objective value vs iterations.